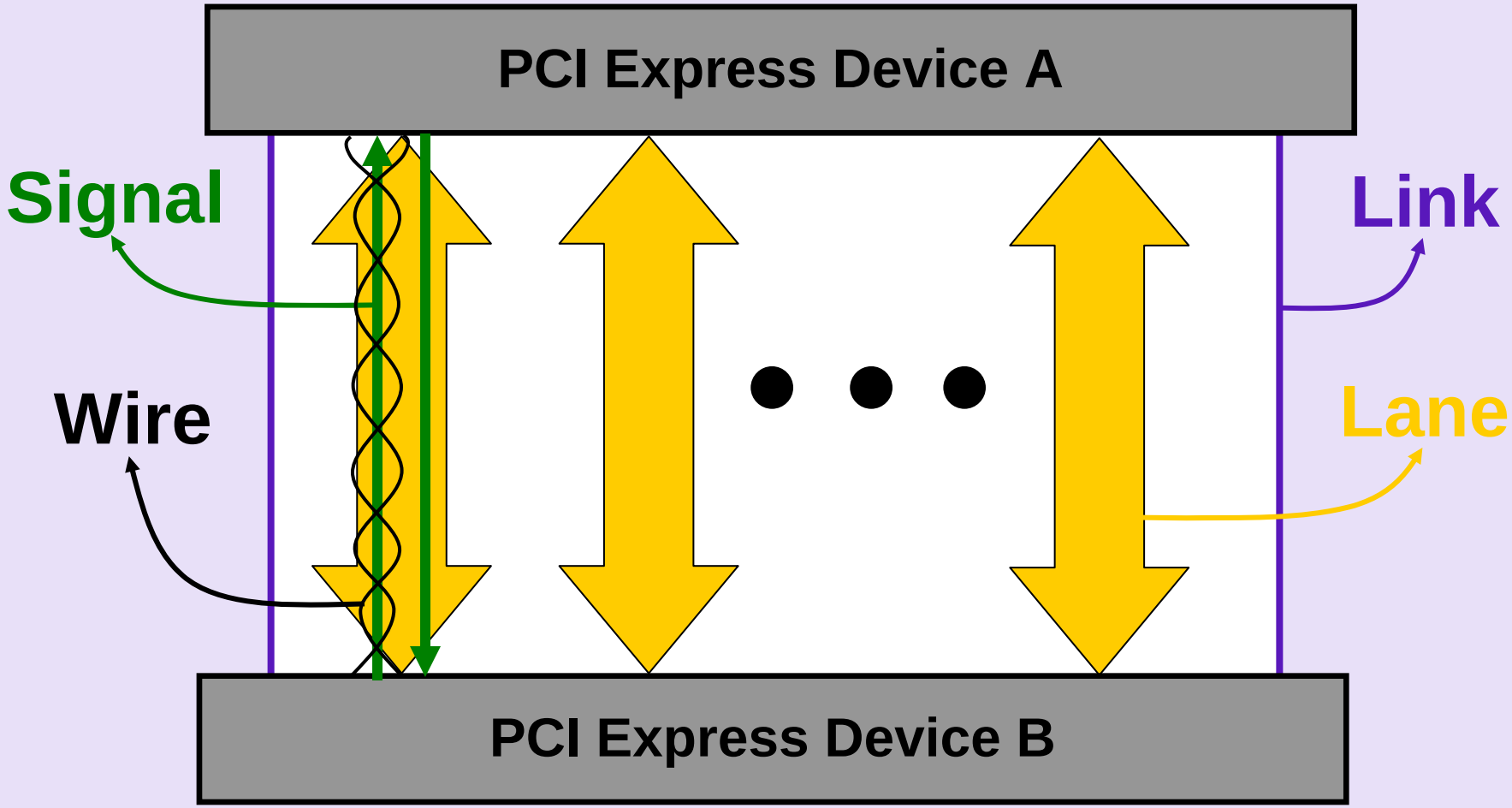
PCI Express® Basics

Ravi Budruk
Senior Staff Engineer and Partner
MindShare, Inc.

PCI Express Introduction

- PCI Express architecture is a high performance, IO interconnect for peripherals in computing/communication platforms
- Evolved from PCI and PCI-X™ architectures
  - ✓ Yet PCI Express architecture is significantly different from its predecessors PCI and PCI-X
- PCI Express is a serial point-to-point interconnect between two devices
- Implements packet based protocol for information transfer
- Scalable performance based on number of signal Lanes implemented on the PCI Express interconnect

PCI Express Terminology



PCI Express Throughput

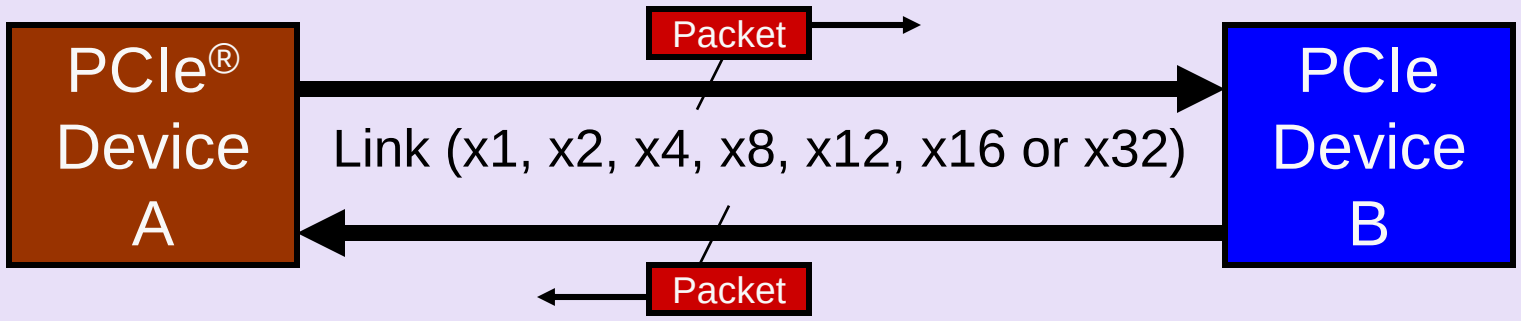
Link Width	x1	x2	x4	x8	x12	x16	x32
Aggregate BW (GBytes/s)	0.5	1	2	4	6	8	16

- Assumes 2.5 GT/s signaling in each direction
- 80% BW utilized due to 8b/10b encoding overhead
- Aggregate bandwidth implies simultaneous traffic in both directions
- Peak bandwidth is higher than any bus available

- PCIe 2.0 PHYs may optionally support 5 GT/s signaling, thus doubling above bandwidth numbers

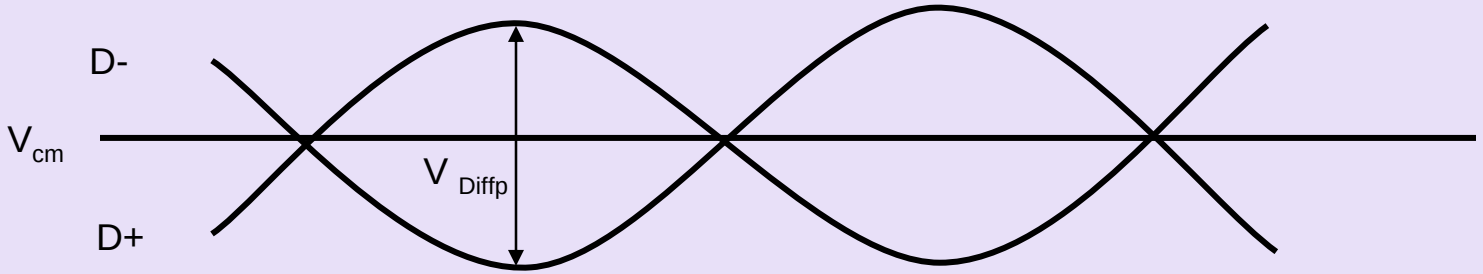
PCI Express Features

- Point-to-point connection
- Serial bus means fewer pins
- Scaleable: x1, x2, x4, x8, x12, x16, x32
- Dual Simplex connection
- 2.5VGT/s transfer/direction/s
- Packet based transaction protocol



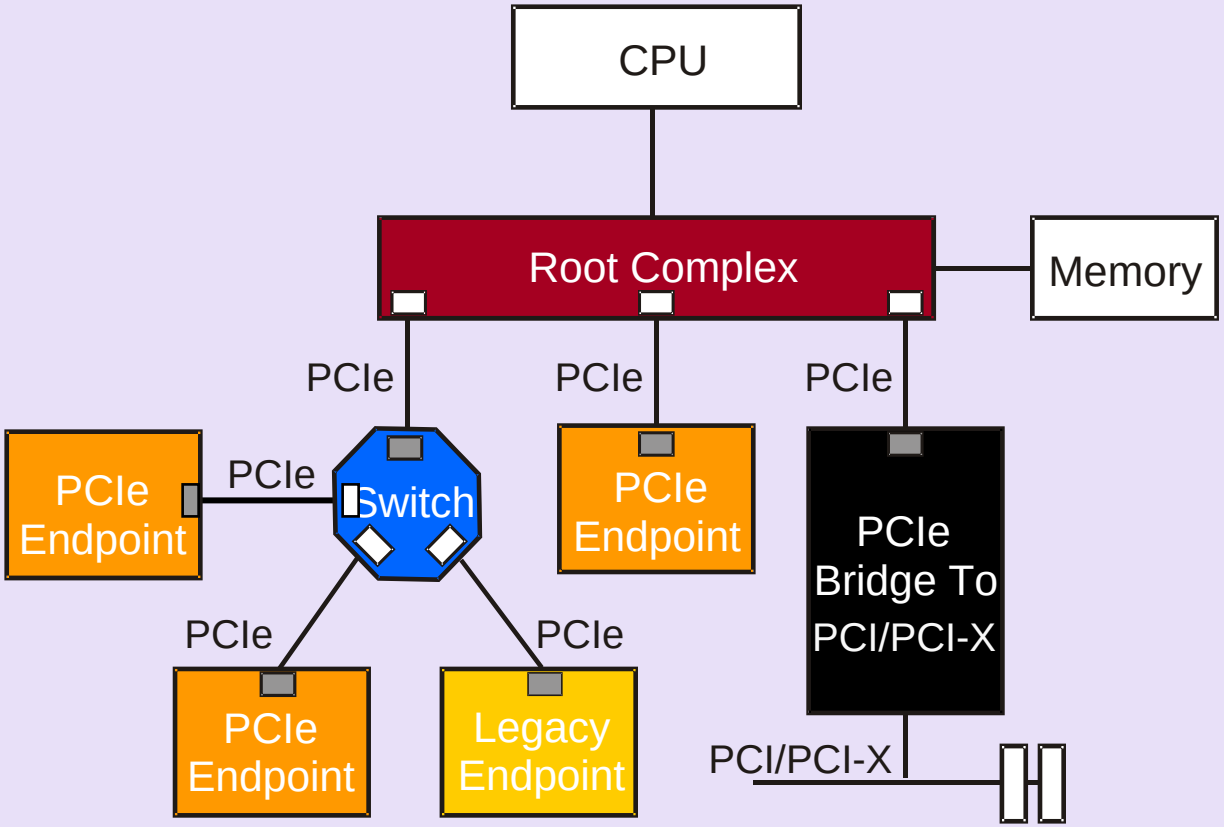
Differential Signaling

- Electrical characteristics of PCI Express signal
  - ✓ Differential signaling
    - Transmitter Differential Peak voltage = 0.4 - 0.6 V
    - Transmitter Common mode voltage = 0 - 3.6 V

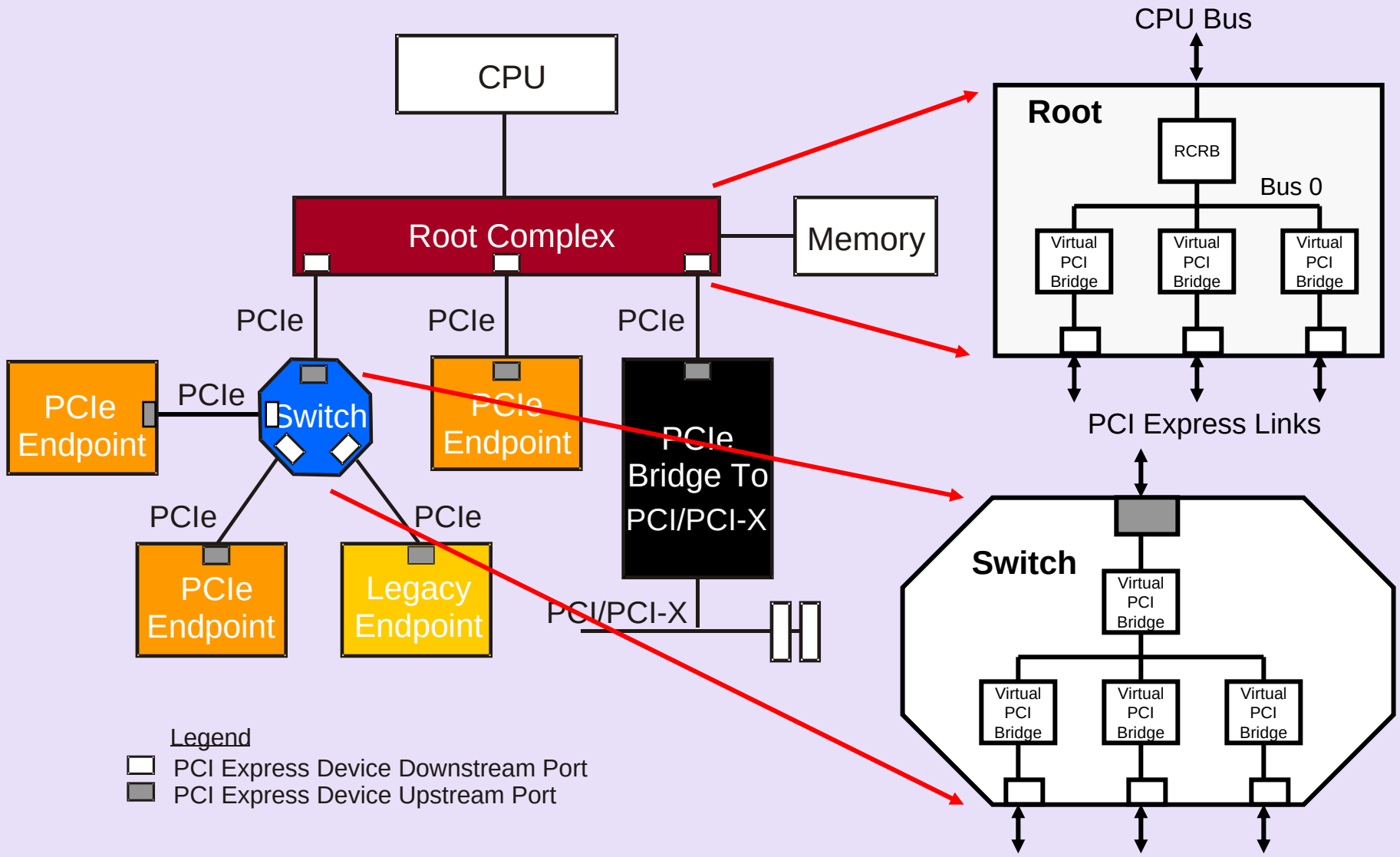


- Two devices at opposite ends of a Link may support different DC common mode voltages

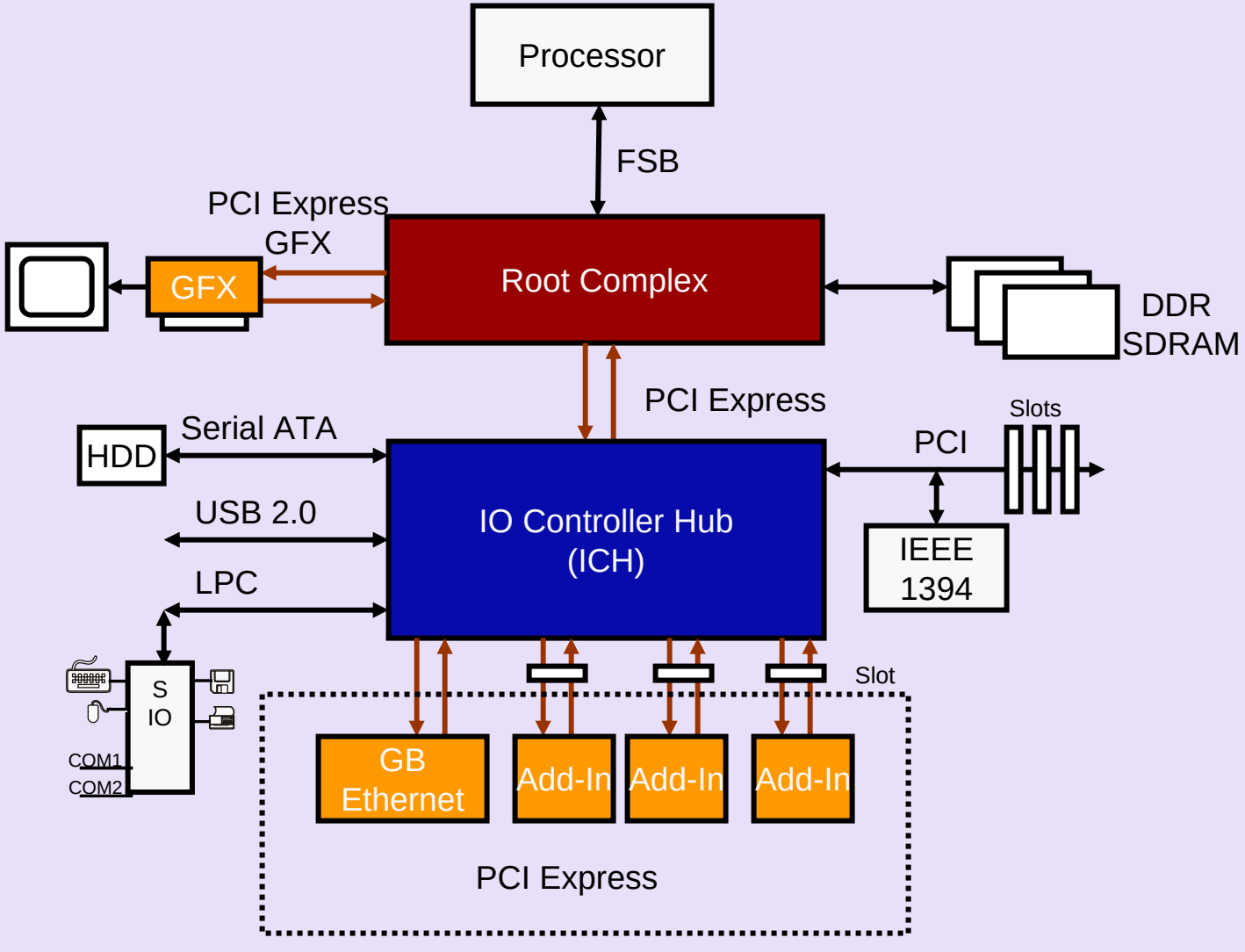
Example PCI Express Topology



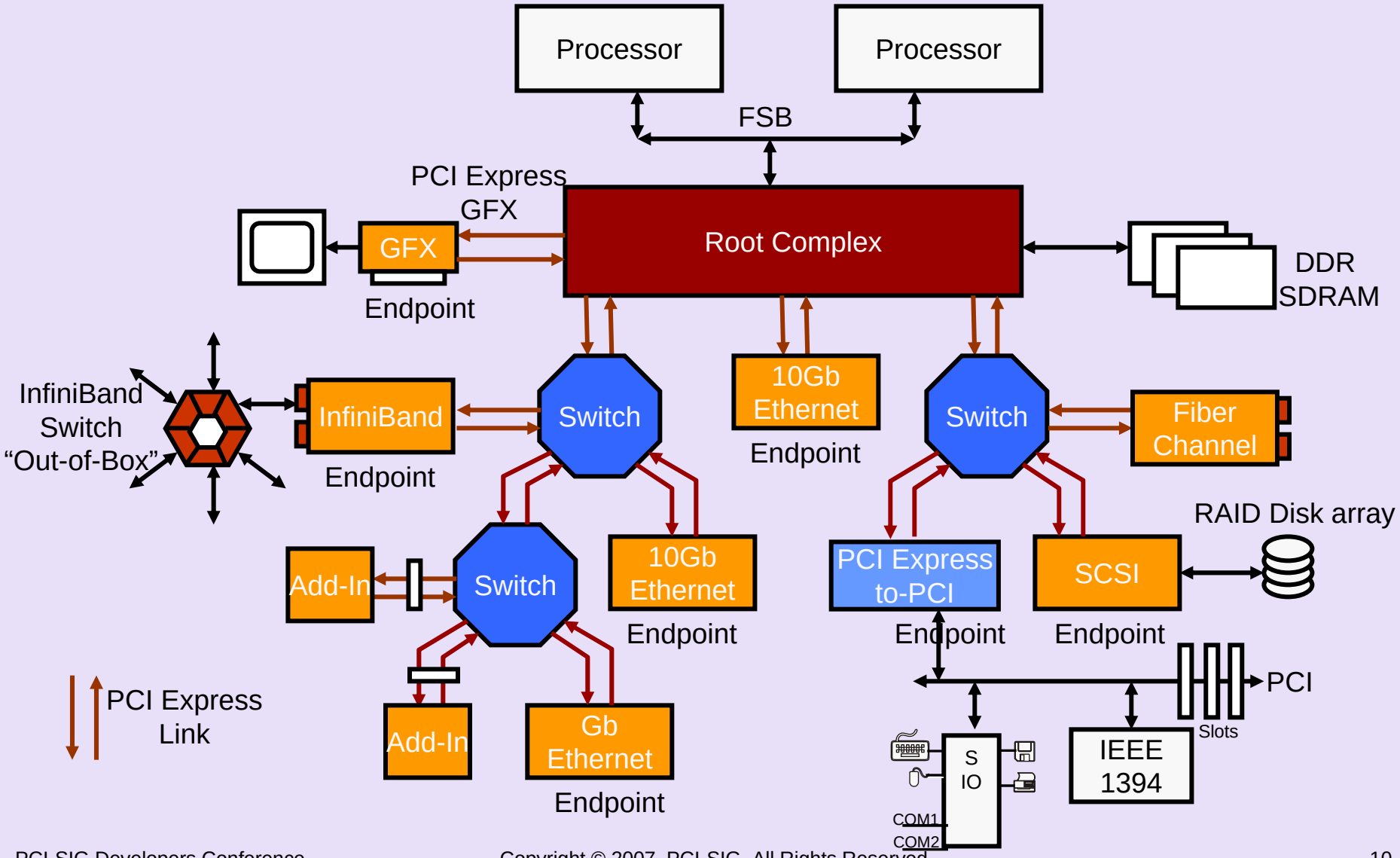
Example PCI Express Topology – Root & Switch



Example Low Cost PCI Express System



Example PCI Express Server System



Transaction Types, Address Spaces

- Request are translated to one of four transaction types by the Transaction Layer:
  1. **Memory Read or Memory Write**. Used to transfer data from or to a memory mapped location
     - The protocol also supports a *locked memory read* transaction variant.
  2. **I/O Read or I/O Write.** Used to transfer data from or to an I/O location
     - These transactions are restricted to supporting legacy endpoint devices.
  3. **Configuration Read or Configuration Write**. Used to discover device capabilities, program features, and check status in the 4KB PCI Express configuration space.
  4. **Messages.** Handled like posted writes. Used for event signaling and general purpose messaging.

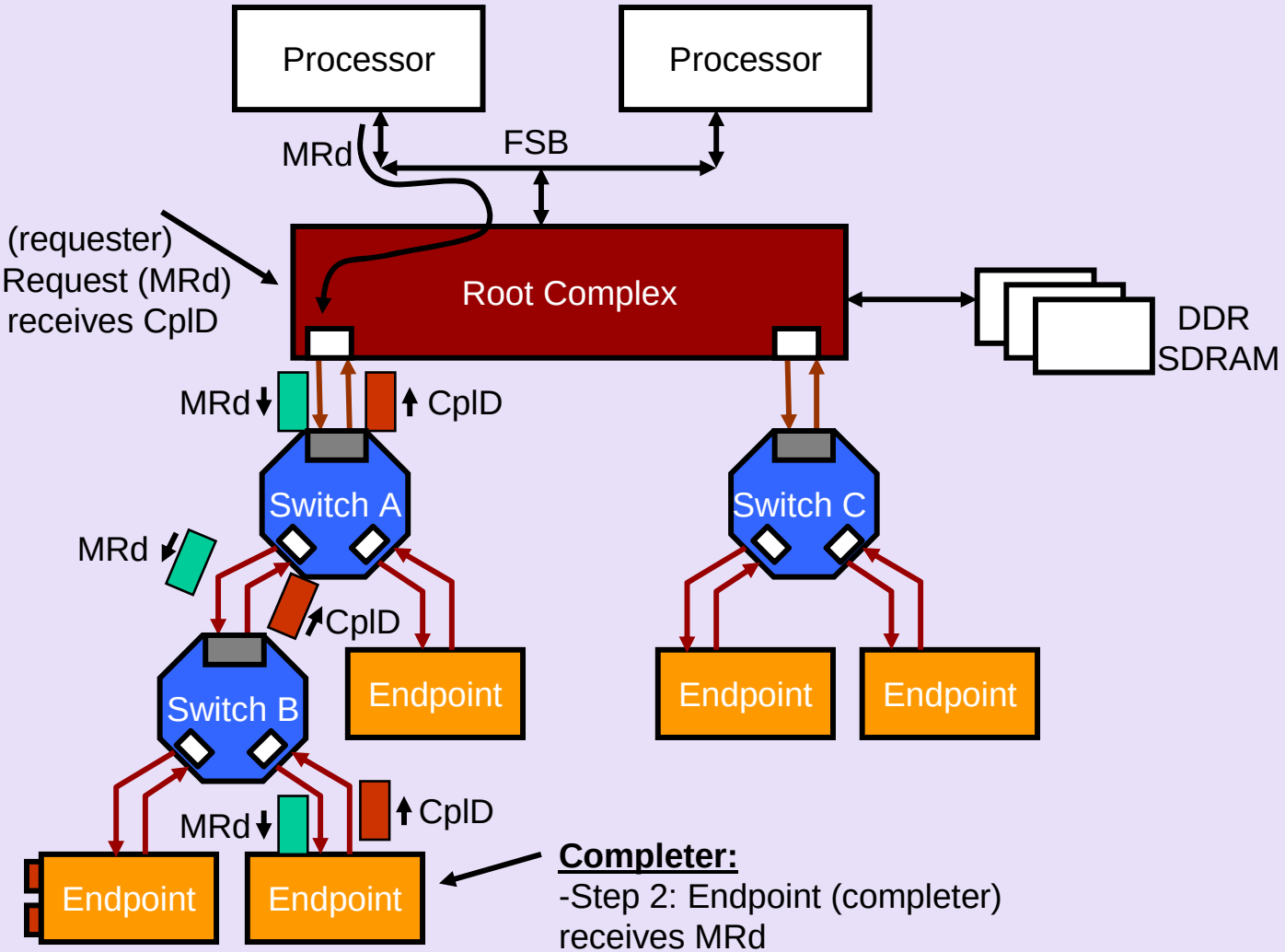
PCI Express TLP Types

Description	Abbreviated Name
Memory Read Request	**MRd**
Memory Read Request – Locked Access	**MRdLk**
Memory Write Request	**MWr**
IO Read Request	**IORd**
IO Write Request	**IOWr**
Configuration Read Request Type 0 and Type 1	**CfgRd0, CfgRd1**
Configuration Write Request Type 0 and Type 1	**CfgWr0, CfgWr1**
Message Request without Data Payload	**Msg**
Message Request with Data Payload	**MsgD**
Completion without Data (used for IO, configuration write completions and read completion with error completion status)	**Cpl**
Completion with Data (used for memory, IO and configuration read completions)	**CplD**
Completion for Locked Memory Read without Data (used for error status)	**CplLk**
Completion for Locked Memory Read with Data	**CplDLk**

Three Methods For Packet Routing

- Each request or completion header is tagged as to its *type*, and each of the packet types is routed based on one of three schemes:
  - ✓ Address Routing
  - ✓ ID Routing
  - ✓ Implicit Routing
- Memory and IO requests use address routing.
- Completions and Configuration cycles use ID routing.
- Message requests have selectable routing based on a 3-bit code in the message routing sub-field of the header type field.

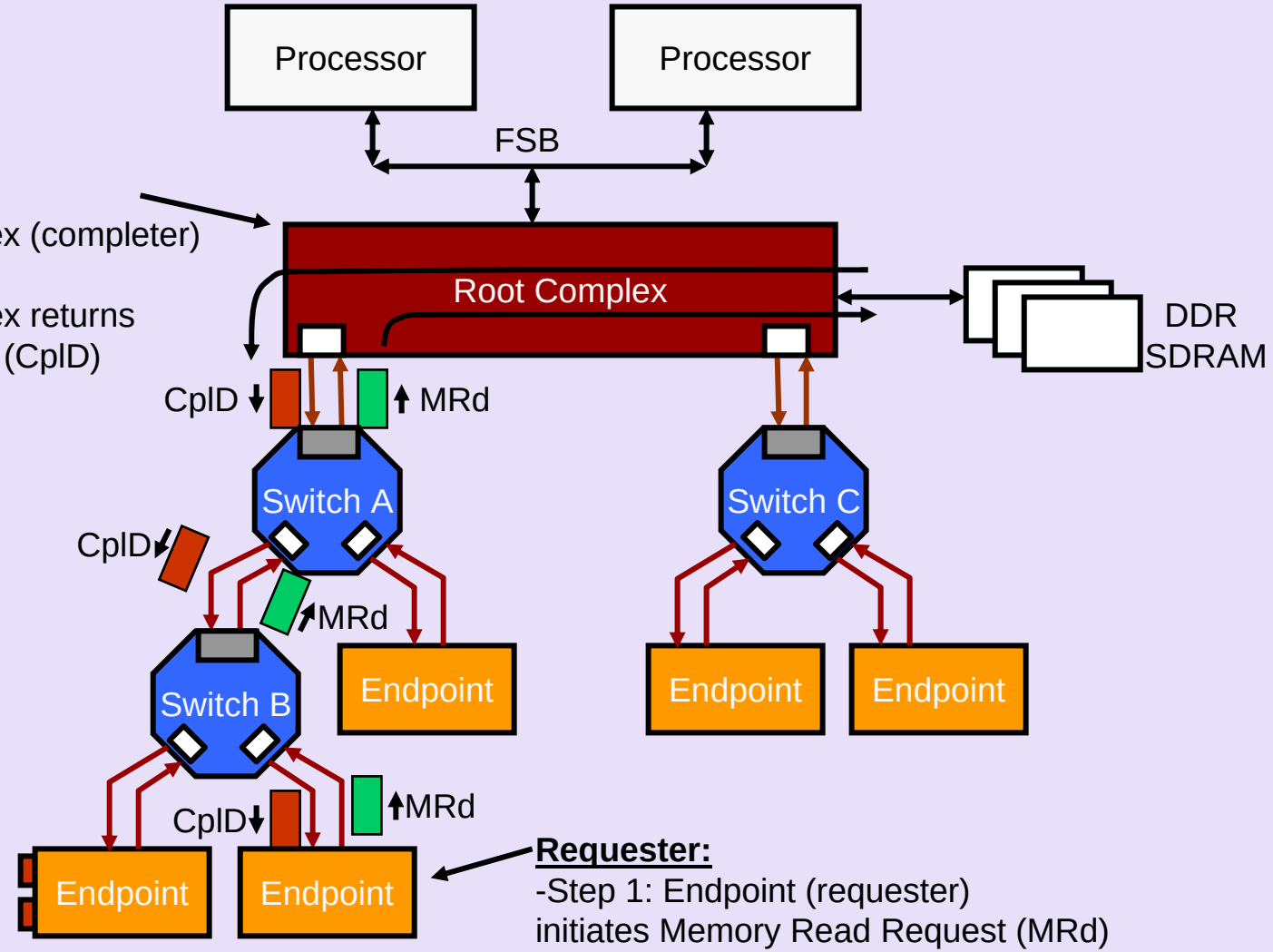
Programmed I/O Transaction



Requester:
-Step 1: Root Complex (requester) initiates Memory Read Request (MRd)
-Step 4: Root Complex receives CplD

Completer:
-Step 2: Endpoint (completer) receives MRd
-Step 3: Endpoint returns Completion with data (CplD)

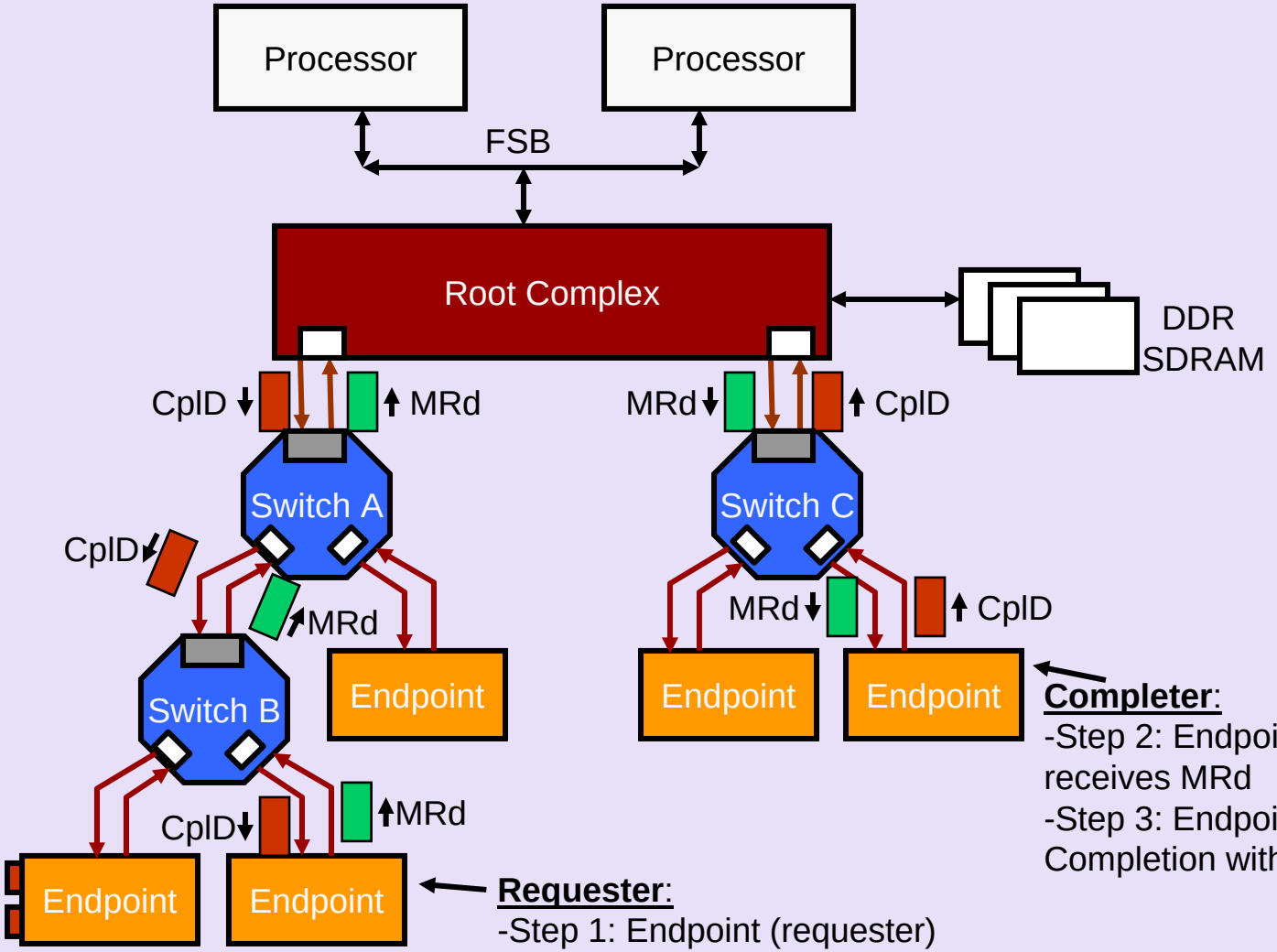
DMA Transaction



Completer:
-Step 2: Root Complex (completer) receives MRd
-Step 3: Root Complex returns Completion with data (CplD)

Requester:
-Step 1: Endpoint (requester) initiates Memory Read Request (MRd)
-Step 4: Endpoint receives CplD

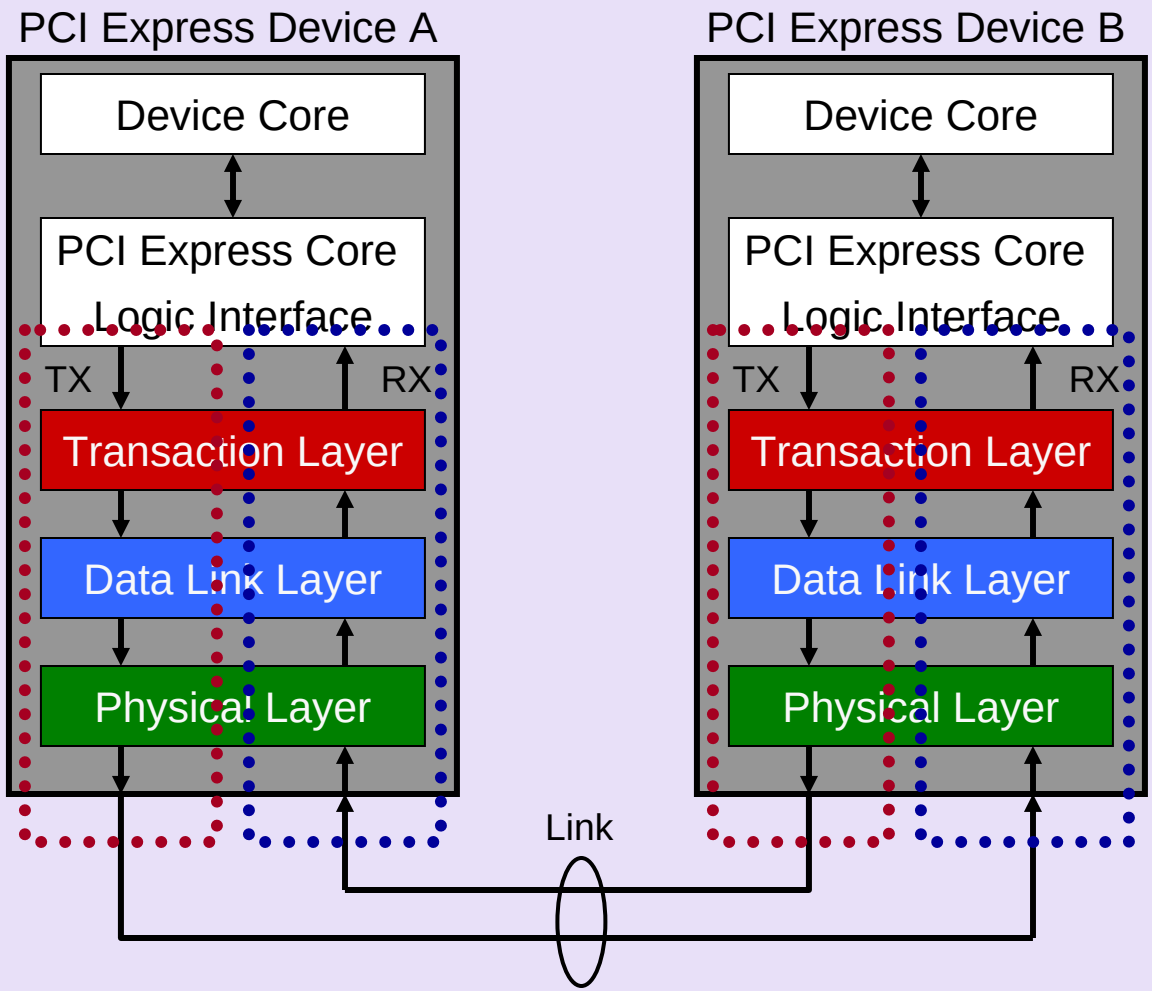
Peer-to-Peer Transaction



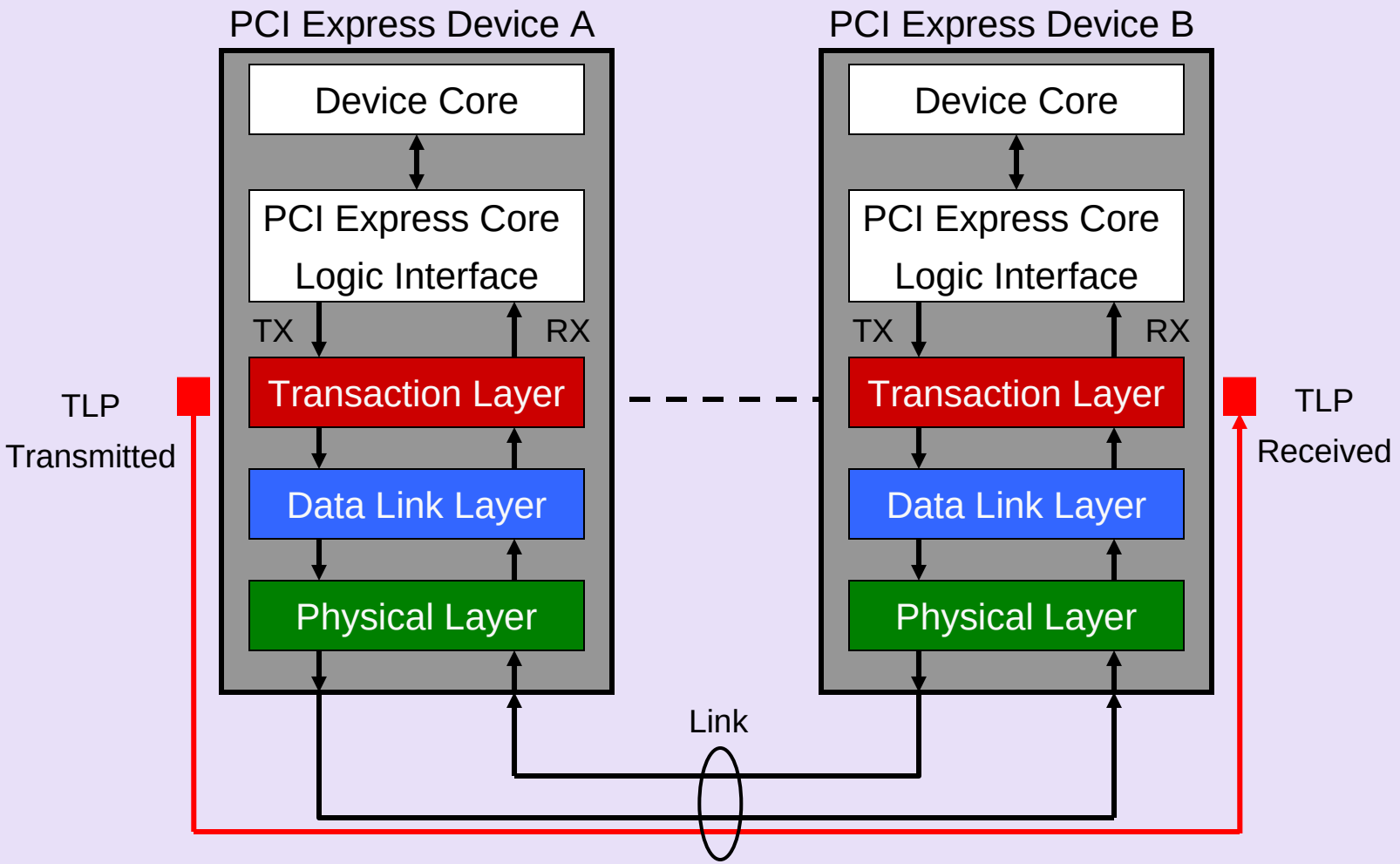
Completer:
-Step 2: Endpoint (completer) receives MRd
-Step 3: Endpoint returns Completion with data (CplD)

Requester:
-Step 1: Endpoint (requester) initiates Memory Read Request (MRd)
-Step 4: Endpoint receives CplD

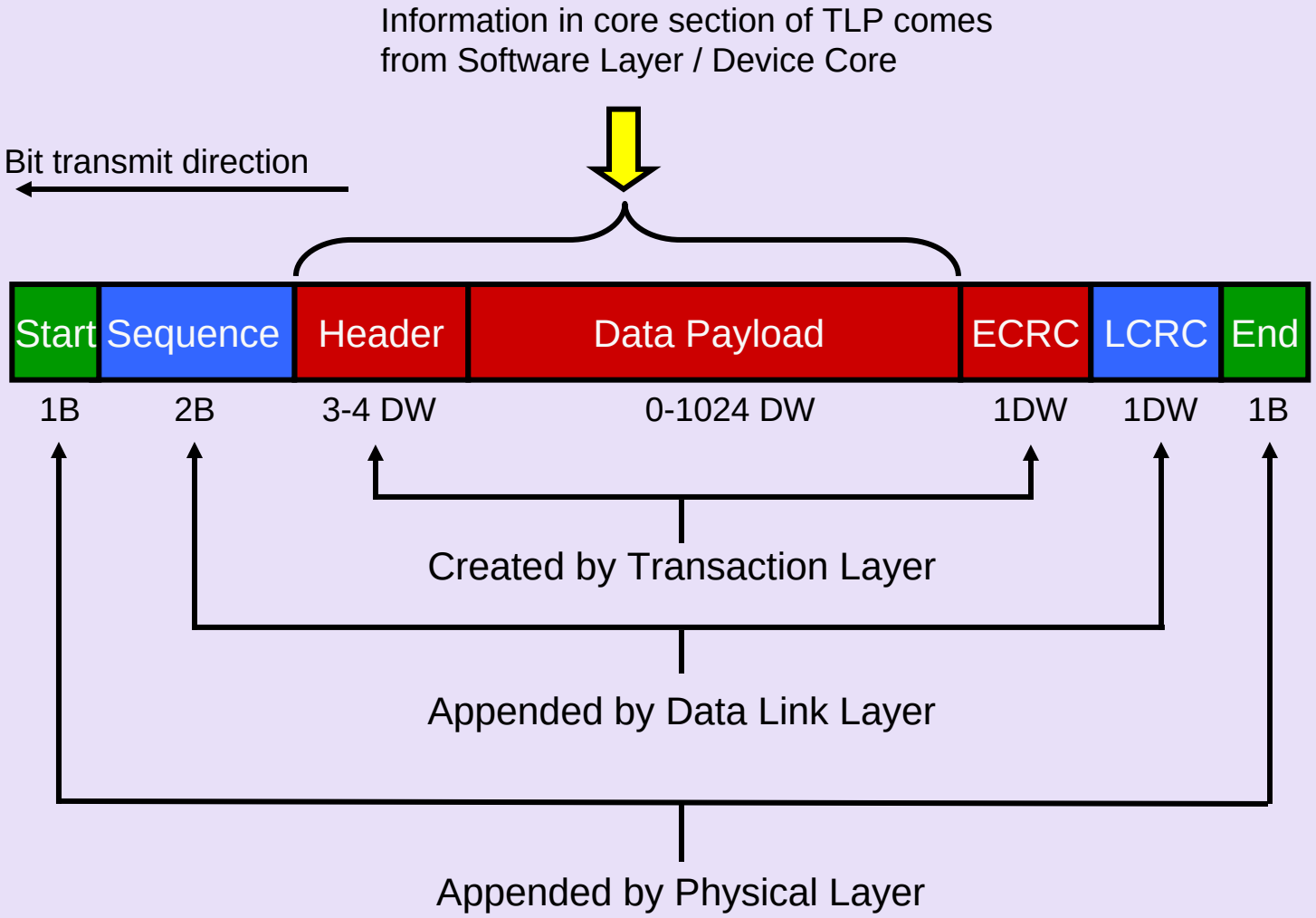
PCI Express Device Layers



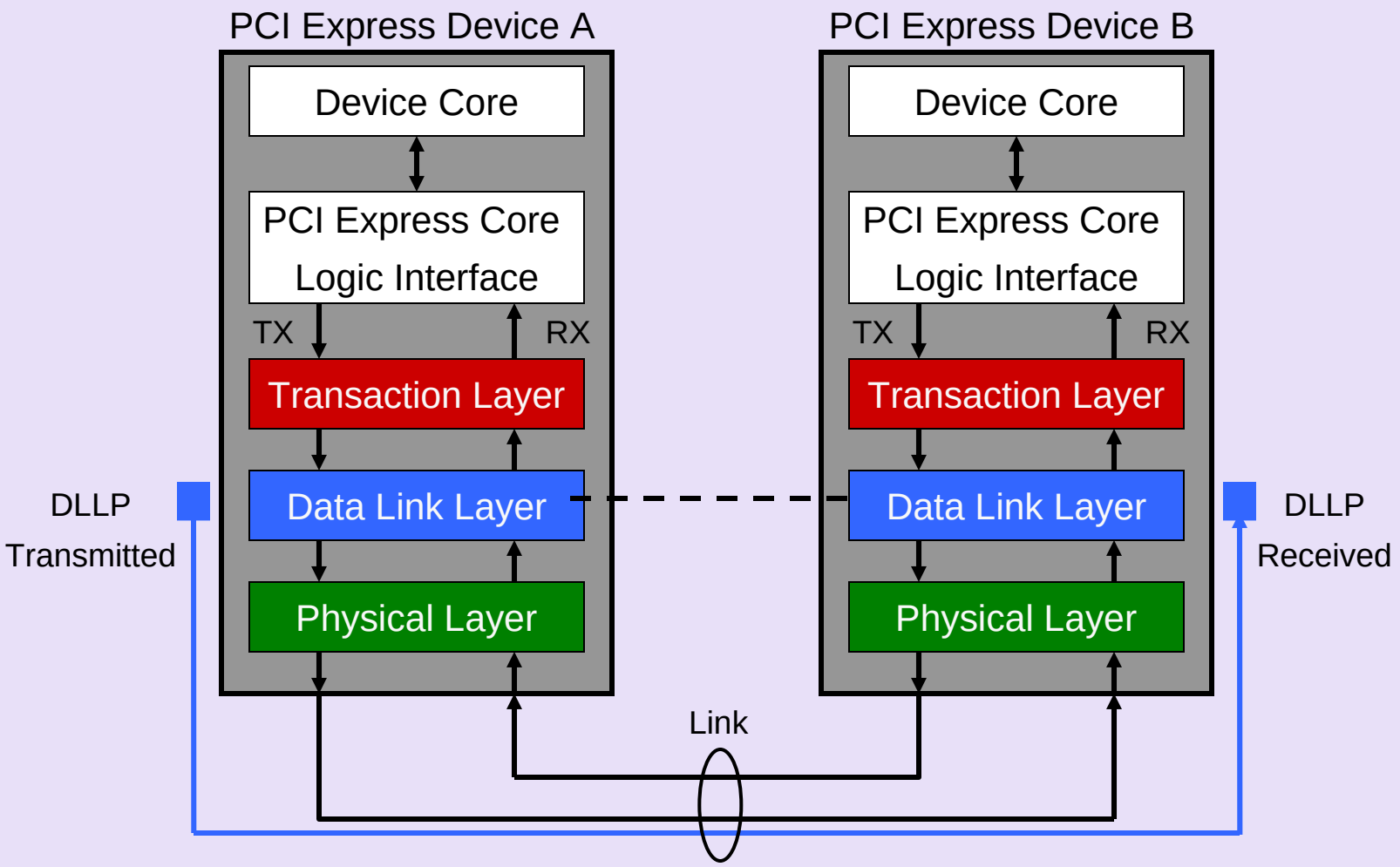
TLP Origin and Destination



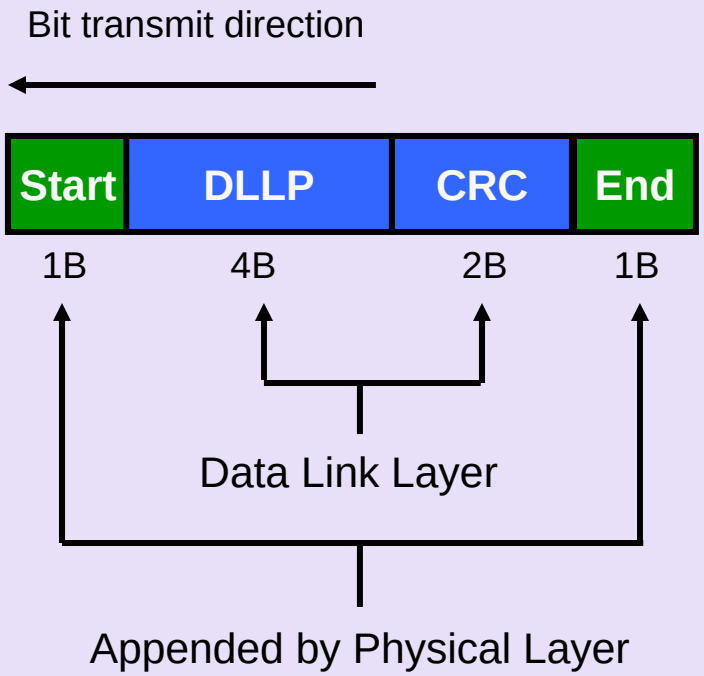
TLP Structure



DLLP Origin and Destination

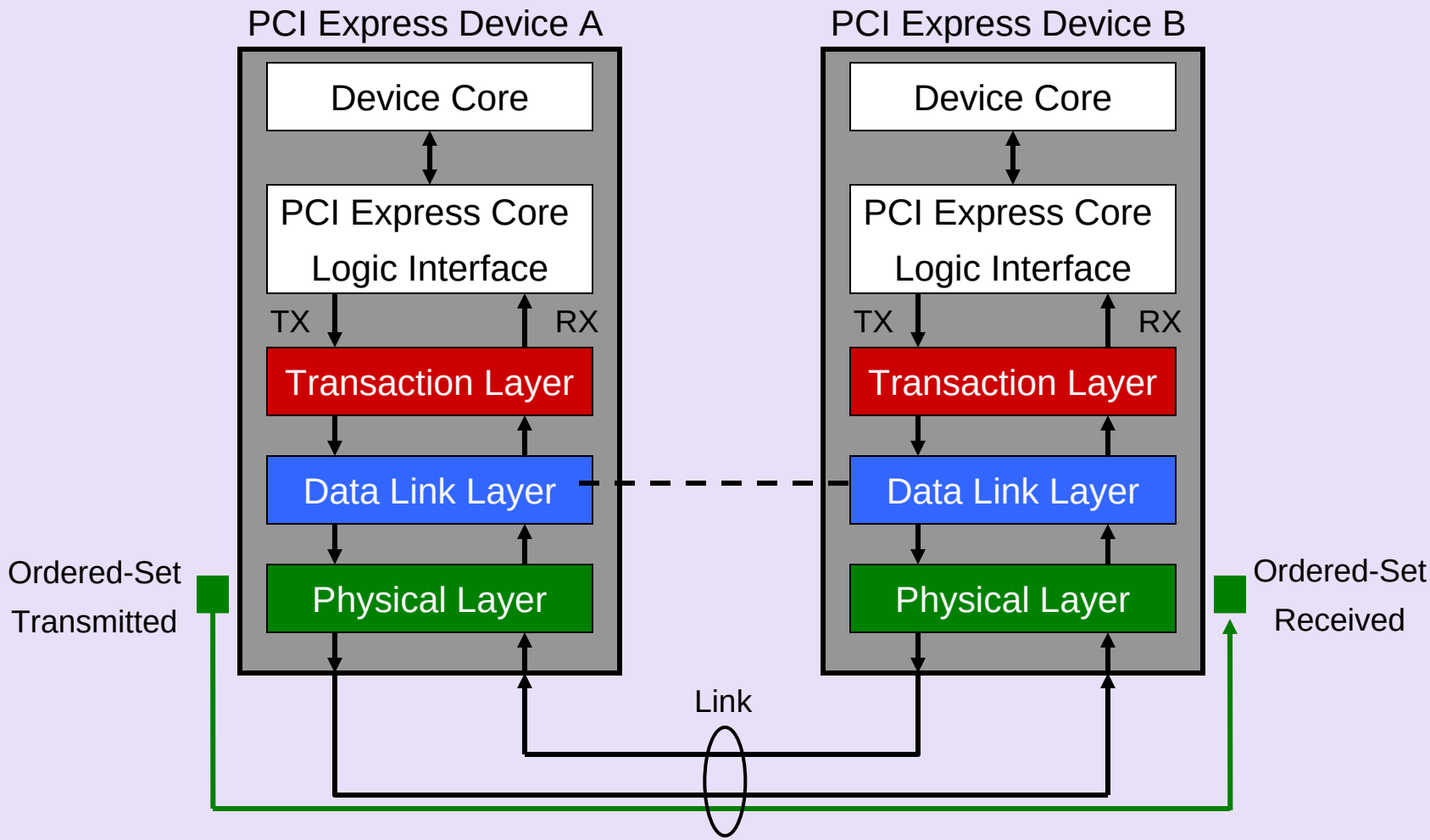


DLLP Structure



- ACK / NAK Packets
- Flow Control Packets
- Power Management Packets
- Vendor Defined Packets

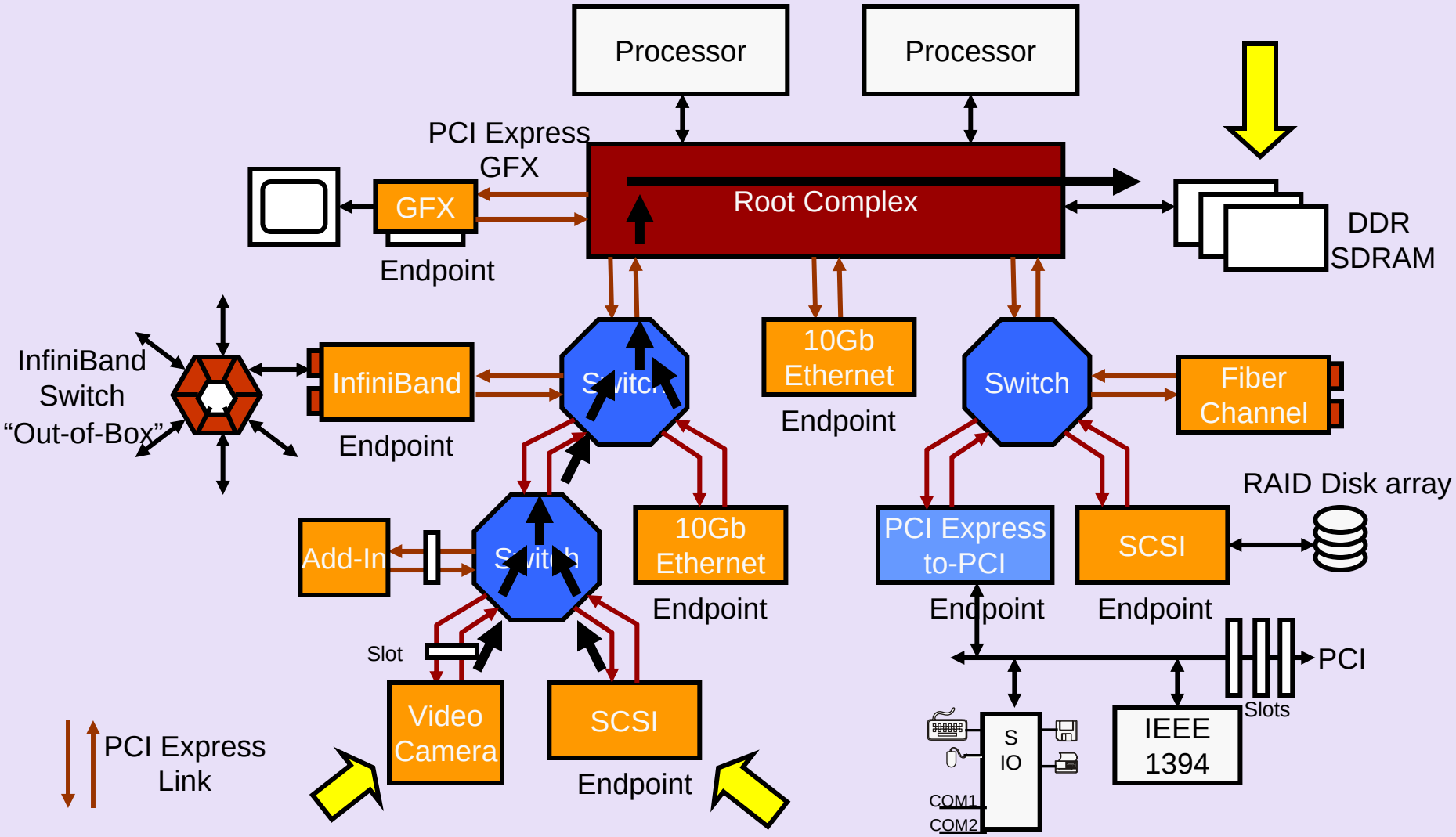
Ordered-Set Origin and Destination



Ordered-Set Structure

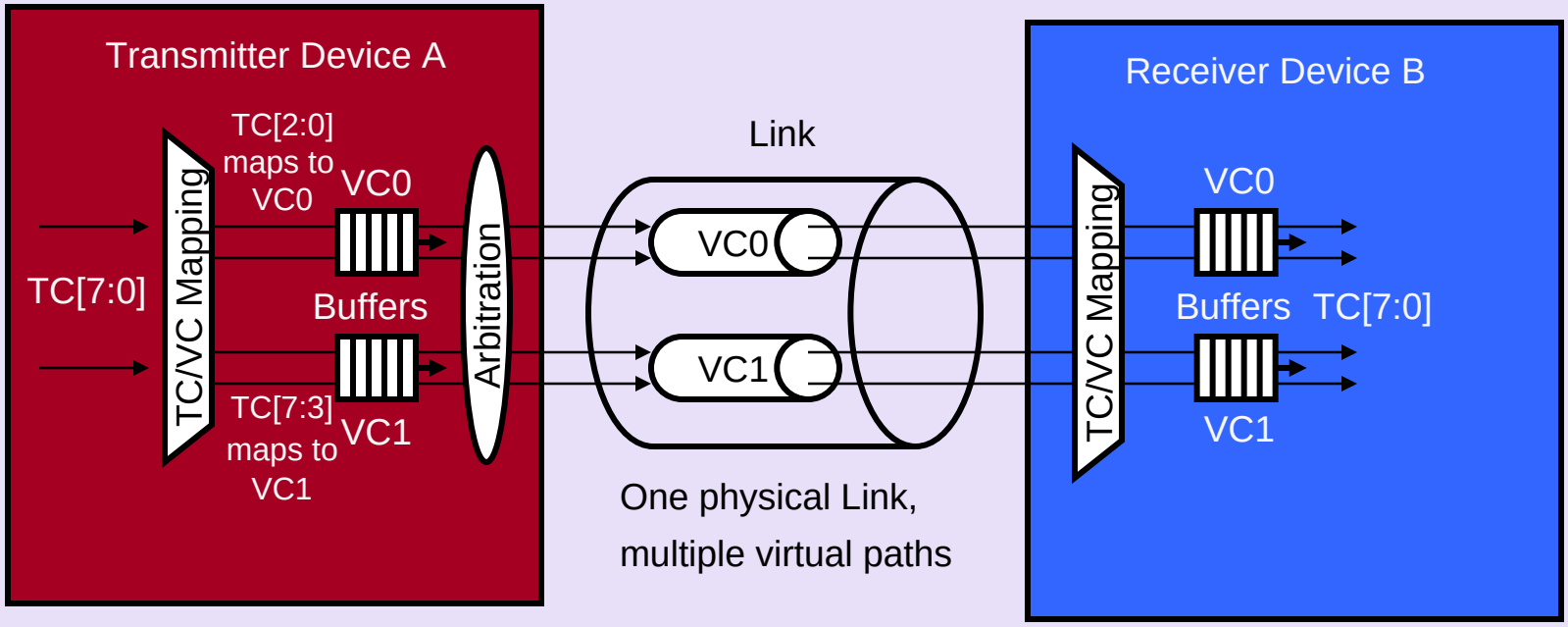
- Training Sequence One (TS1)
  - ✓ 16 character set: 1 COM, 15 TS1 data characters
- Training Sequence Two (TS2)
  - ✓ 16 character set: 1 COM, 15 TS2 data characters
- SKIP
  - ✓ 4 character set: 1 COM followed by 3 SKP identifiers
- Electrical Idle (IDLE)
  - ✓ 4 characters: 1 COM followed by 3 IDL identifiers
- Fast Training Sequence (FTS)
  - ✓ 4 characters: 1 COM followed by 3 FTS identifiers

Quality of Service

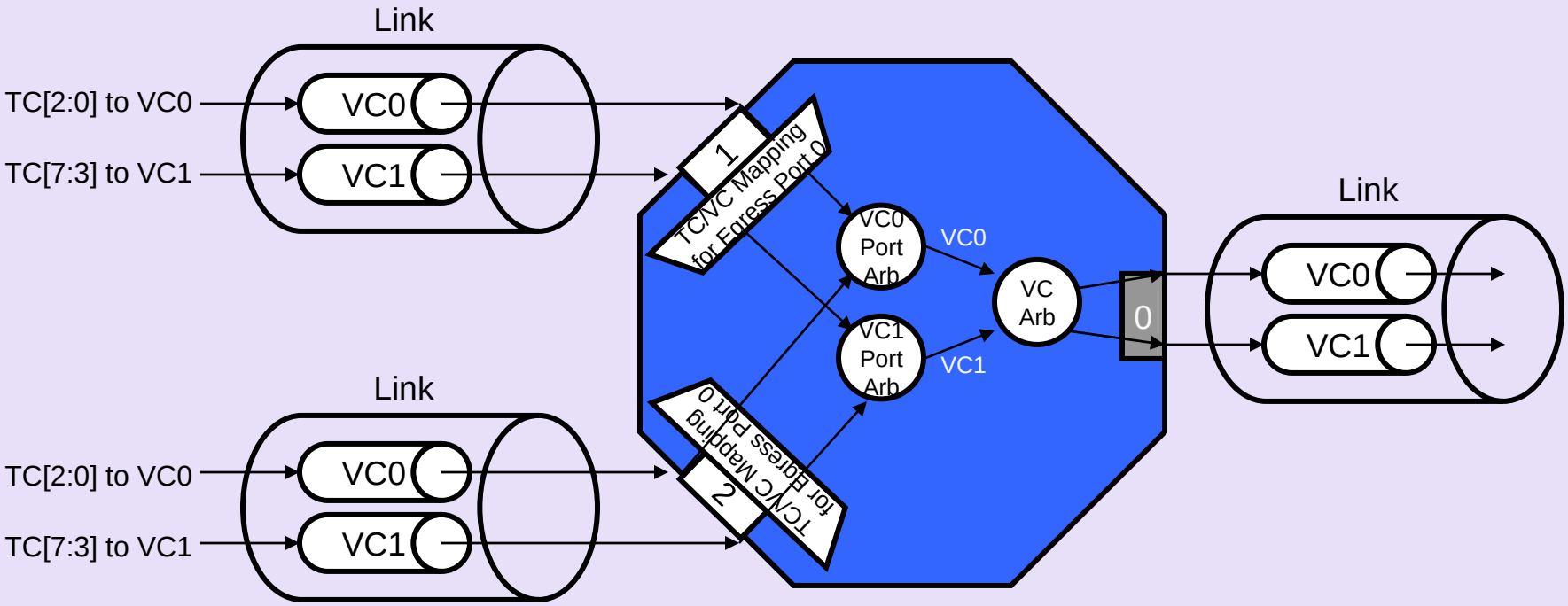


Traffic Classes and Virtual Channels

- Quality of Service (QoS) policy through Virtual Channel and Traffic Class tags

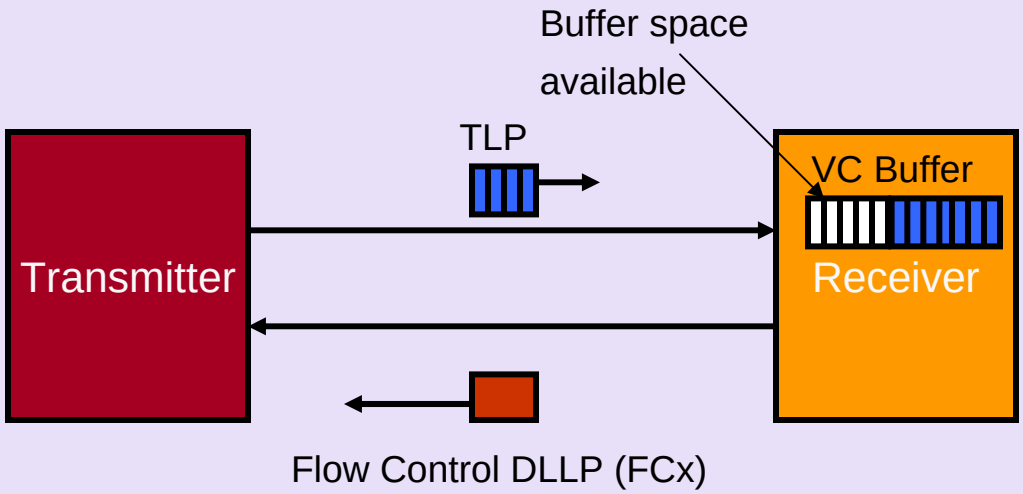


Port Arbitration and VC Arbitration



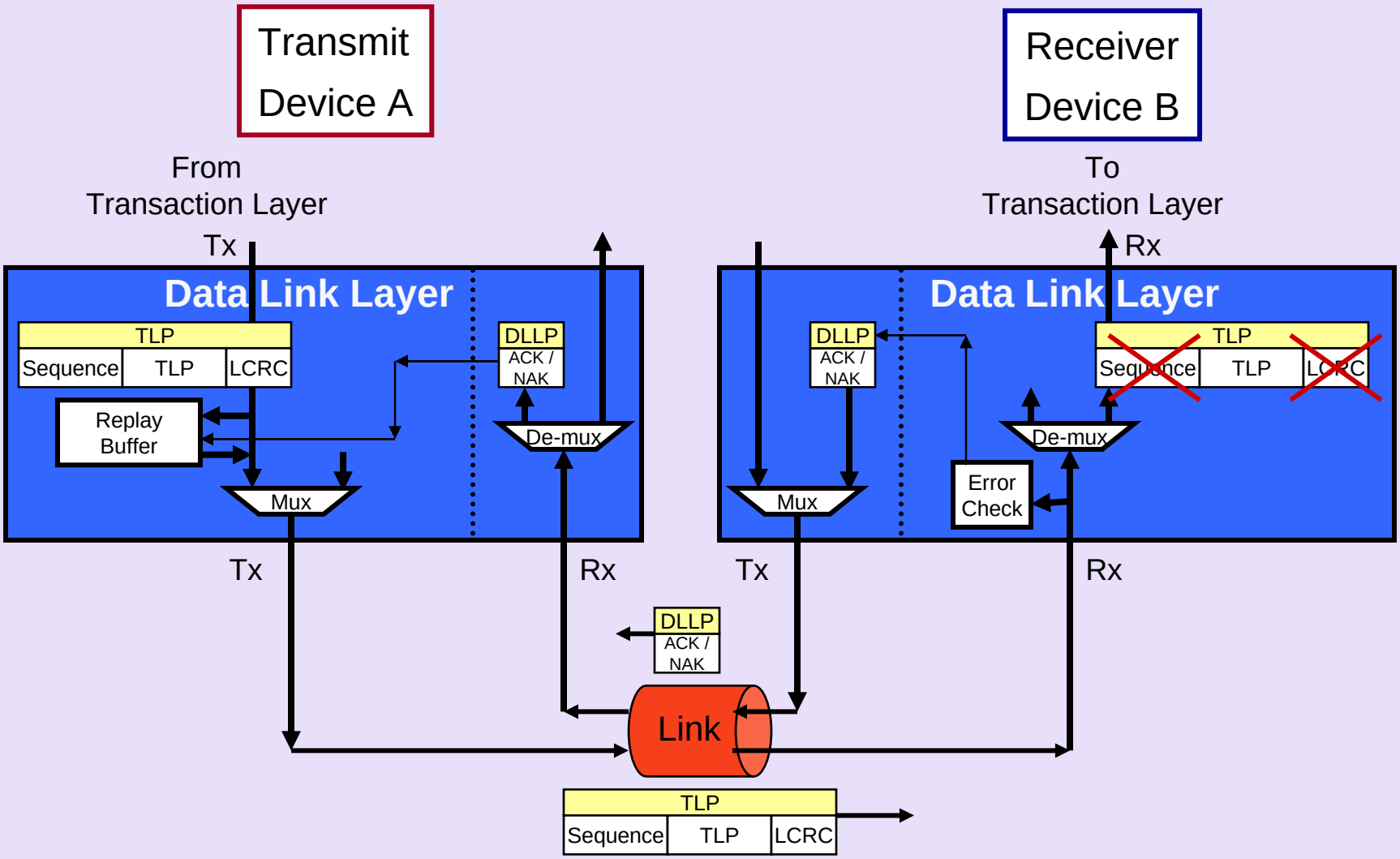
PCI Express Flow Control

- Credit-based *flow control* is point-to-point based, not end-to-end

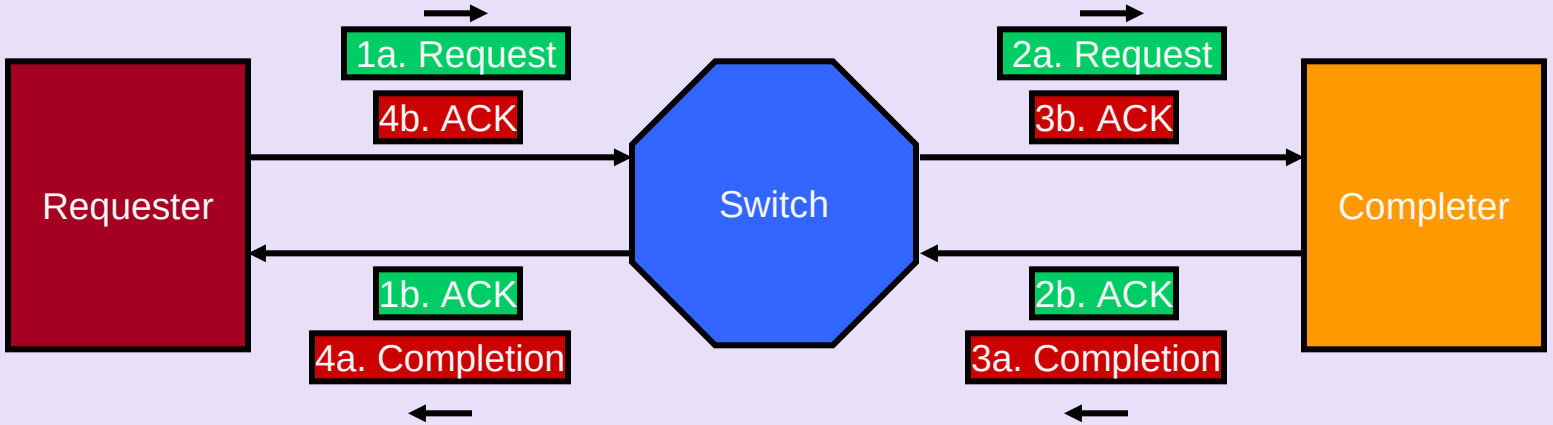


Receiver sends Flow Control Packets (FCP) which are a type of DLLP (Data Link Layer Packet) to provide the transmitter with credits so that it can transmit packets to the receiver

ACK/NAK Protocol Overview



ACK/NAK Protocol: Point-to-Point

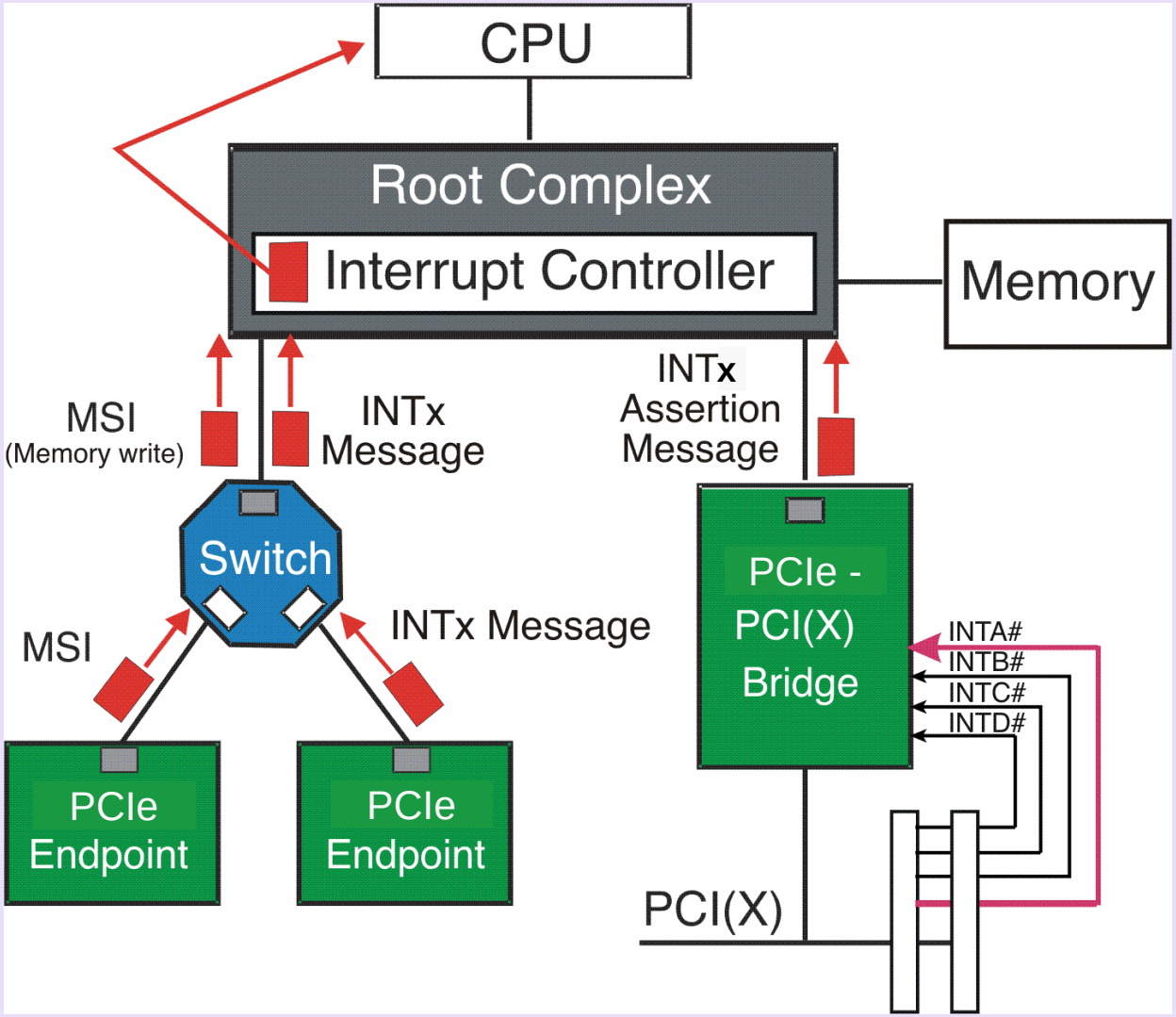


ACK returned for good reception of Request or Completion
NAK returned for error reception of Request or Completion

Interrupt Model: Three Methods

- PCI Express supports three interrupt reporting mechanisms:
  1. **Message Signaled Interrupts (MSI)**
     - Legacy endpoints are required to support MSI (or MSI-X) with 32- or 64-bit MSI capability register implementation
     - Native PCI Express endpoints are required to support MSI with 64-bit MSI capability register implementation
  2. **Message Signaled Interrupts - X (MSI-X)**
     - Legacy and native endpoints are required to support MSI-X (or MSI) and implement the associated MSI-X capability register
  3. **INTx Emulation**.
     - Native and Legacy endpoints are required to support Legacy INTx Emulation
     - PCI Express defines in-band messages which emulate the four physical interrupt signals (INTA-INTD) routed between PCI devices and the system interrupt controller
     - Forwarding support required by switches

Native and Legacy Interrupts



PCI Express Error Handling

- All PCI Express devices are required to support some combination of:
  - ✓ Existing software written for generic PCI error handling, and which takes advantage of the fact that PCI Express has mapped many of its error conditions to existing PCI error handling mechanisms.
  - ✓ Additional PCI Express-specific reporting mechanisms
- Errors are classified as ***correctable*** and ***uncorrectable***.
- *Uncorrectable* errors are further divided into:
  - ✓ Fatal uncorrectable errors
  - ✓ Non-fatal uncorrectable errors.

Correctable Errors

- Errors classified as correctable, degrade system performance, but recovery can occur with no loss of information
  - ✓ Hardware is responsible for recovery from a correctable error and no software intervention is required.
- Even though hardware handles the correction, logging the frequency of correctable errors may be useful if software is monitoring link operations.
- An example of a correctable error is the detection of a link CRC (LCRC) error when a TLP is sent, resulting in a Data Link Layer retry event.

Uncorrectable Errors

- Errors classified as uncorrectable impair the functionality of the interface and there is no specification mechanism to correct these errors
- The two subgroups are fatal and non-fatal
  1. **Fatal Uncorrectable Errors**: Errors which render the link unreliable
     - First-level strategy for recovery may involve a link reset by the system
     - Handling of fatal errors is platform-specific
  2. **Non-Fatal Uncorrectable Errors**: Uncorrectable errors associated with a particular transaction, while the link itself is reliable
     - Software may limit recovery strategy to the device(s) involved
     - Transactions between other devices are not affected

Baseline Error Reporting

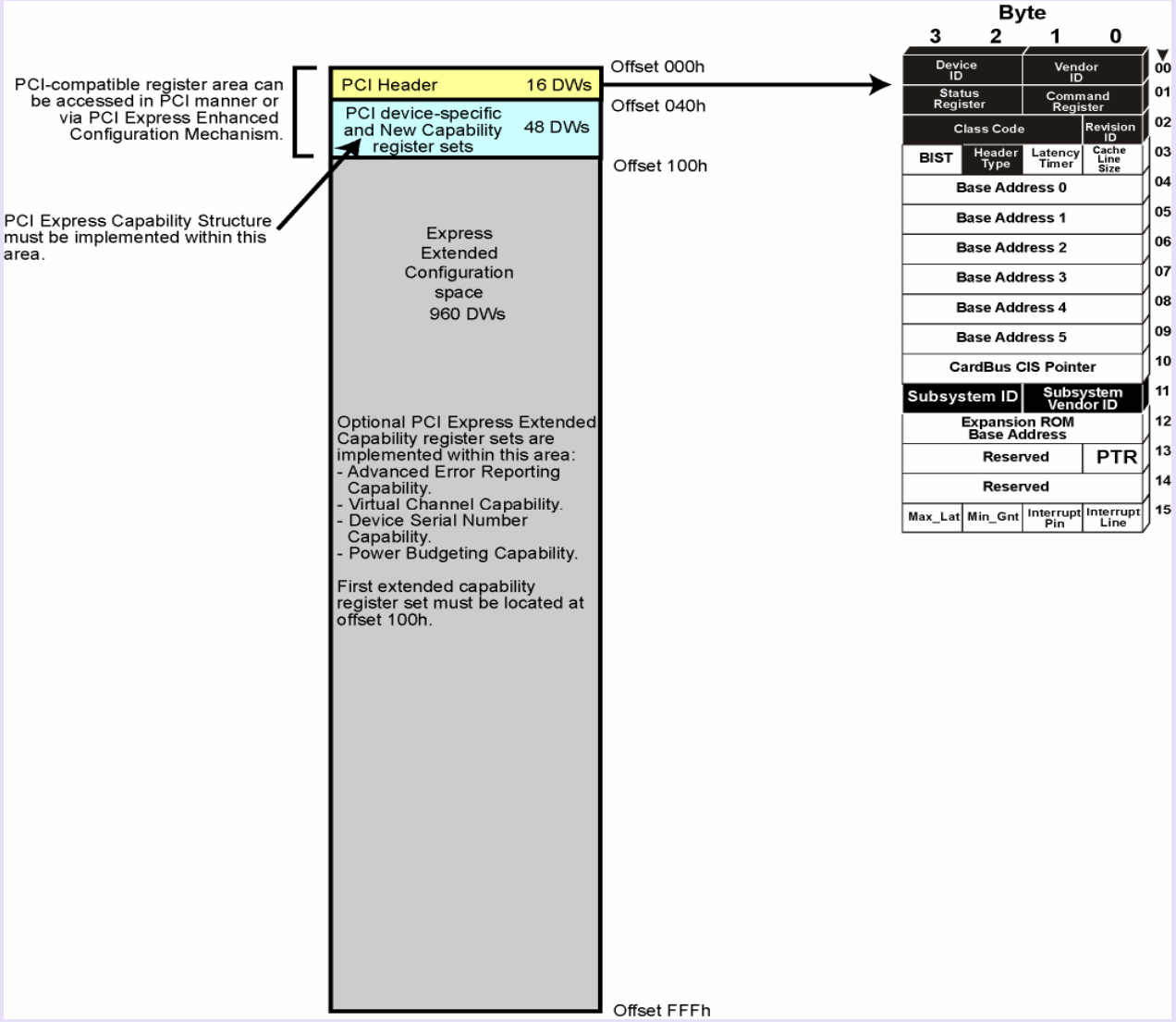
- Enabling/disabling error reporting
- Providing error status
- Providing error status for Link Training
- Initiating Link Re-training

- Registers provide control and status for
  - ✓ Correctable errors
  - ✓ Non-fatal uncorrectable errors
  - ✓ Fatal uncorrectable errors
  - ✓ Unsupported request errors

Advanced Error Reporting

- Finer granularity in defining error type
- Ability to define severity of uncorrectable errors
  - ✓ Either send ERR_FATAL or ERR_NONFATAL message for a given error
- Support for error logging of error type and TLP header related to error
- Ability to mask reporting of errors
- Enable/disable root reporting of errors
- Identify source of errors

PCI Express Configuration Space



Summary of Changes for 2.0

- Higher speed (5.0 GT/s), supported by:
  - ✓ Selectable de-emphasis levels
  - ✓ Selectable transmitter voltage range
- Dynamic speed and link width changes
  - ✓ Power savings, higher bandwidth, reliability
- Virtualization support
  - ✓ Access Control Services
- Other New Features
  - ✓ Completion timeout control
  - ✓ Function Level Reset
  - ✓ Modified Compliance Pattern for testing

Thank you for attending the
PCI-SIG Developers Conference 2007.

For more information please go to
www.pcisig.com

PCI Express Basics

Ravi Budruk
Senior Staff Engineer and Partner
MindShare, Inc.